

## Глава 3

# ОСНОВЫ АЛГОРИТМИЗАЦИИ

### § 3.1

## Алгоритмы и исполнители

*Ключевые слова:*

- алгоритм
- свойства алгоритма
  - ✓ дискретность
  - ✓ понятность
  - ✓ определённость
  - ✓ результативность
  - ✓ массовость
- исполнитель
- характеристики исполнителя
  - ✓ круг решаемых задач
  - ✓ среда
  - ✓ режим работы
  - ✓ система команд
- формальное исполнение алгоритма

#### 3.1.1. Понятие алгоритма

Каждый человек в повседневной жизни, в учёбе или на работе решает огромное количество задач самой разной сложности. Сложные задачи требуют длительных размышлений для нахождения решения; простые и привычные задачи человек решает, не задумываясь, автоматически. В большинстве случаев решение каждой задачи можно разбить на простые этапы (шаги). Для многих таких задач (установка программного обеспечения, сборка шкафа, создание сайта, эксплуатация технического устройства, покупка авиабилета через Интернет и т. д.) уже разработаны и предлагаются пошаговые инструкции, при последовательном выполнении которых можно прийти к желаемому результату.

##### Пример 1

Решение задачи «Зарегистрировать аккаунт в VK» можно осуществить за шесть шагов:

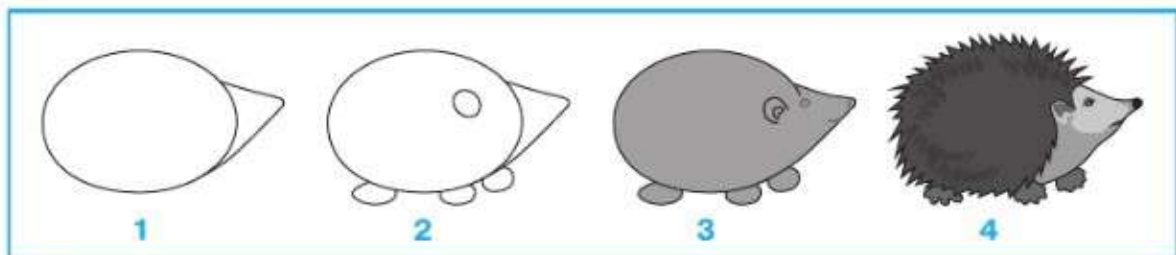
1. Зайти на сайт [vk.com](http://vk.com).
2. Нажать на кнопку **Зарегистрироваться**.



3. Ввести номер телефона.
4. Подтвердить вход (ввести в окно ввода код из СМС).
5. Ввести информацию о себе (имя, фамилия, дата рождения, пол).
6. Подтвердить ввод данных.

### Пример 2

Этапы решения задачи «Нарисовать весёлого ёжика» представлены графически:



### Пример 3

Задача «Найти наибольший общий делитель (НОД) двух целых чисел» может быть решена так:

1. Разложить первое число на простые множители.
2. Разложить второе число на простые множители.
3. Выписать все простые множители, общие для двух чисел.
4. Вычислить НОД путём перемножения общих простых множителей.

### Пример 4

Самое простое решение задачи «Сортировка мусора» может быть таким:

1. Органические отходы (частицы пищи, растения) поместить в бак серого цвета.
2. Неорганические отходы (пластик, бумага, картон, стекло, металл) поместить в бак синего цвета.
3. Ртутные градусники, люминесцентные лампы, батарейки и аккумуляторы сложить отдельно, в небольшую коробку.

Регистрация аккаунта, рисование ёжика, нахождение НОД и сортировка мусора — на первый взгляд, совершенно разные

процессы. Но у них есть общая черта: каждый из этих процессов описывается последовательностью кратких указаний, точное следование которым позволяет получить требуемый результат. Последовательности указаний, приведённые в примерах 1–4, являются алгоритмами решения соответствующих задач. Исполнитель этих алгоритмов — человек.

Алгоритм может представлять собой описание некоторой последовательности вычислений (пример 3) или шагов нематематического характера (примеры 1, 2, 4). Но в любом случае перед его разработкой должны быть чётко определены начальные условия (исходные данные) и то, что предстоит получить (результат). Можно сказать, что **алгоритм** — это описание последовательности шагов в решении задачи, приводящих от исходных данных к требуемому результату.

В общем виде схему работы алгоритма можно представить следующим образом (рис. 3.1).

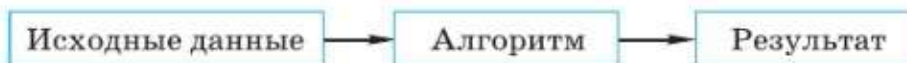


Рис. 3.1. Общая схема работы алгоритма

Алгоритмами являются изучаемые в школе правила сложения, вычитания, умножения и деления чисел, многие грамматические правила, правила геометрических построений и т. д.

### Пример 5

Следующий алгоритм приводит к тому, что из одного натурального числа получается другое натуральное число:

1. Исходное число переводится в двоичную систему счисления.
2. Подсчитывается число единиц в полученной двоичной записи исходного числа.
3. Если число единиц в двоичной записи нечётное, то к этой записи справа приписывается 1, если чётное — 0.
4. Число, соответствующее дополненной двоичной записи, переводится в десятичную систему счисления.

Получившееся таким образом число является результатом работы алгоритма.

Так, если исходным было число 21, то результатом работы алгоритма будет число 43, а если исходным числом было 15, то результатом работы алгоритма будет число 30.

### 3.1.2. Исполнитель алгоритма

Каждый алгоритм предназначен для определённого исполнителя.



**Исполнитель** — это некоторый объект (человек, животное, техническое устройство), способный выполнять определённый набор команд.

Различают формальных и неформальных исполнителей. Формальный исполнитель не вникает в смысл того, что он делает, и не рассуждает, почему он поступает так, а не иначе. Одну и ту же команду формальный исполнитель всегда выполняет одинаково. Неформальный исполнитель может выполнять команду по-разному.

Рассмотрим более подробно множество формальных исполнителей. Формальные исполнители необычайно разнообразны, но для каждого из них можно указать следующие характеристики: круг решаемых задач (назначение), среду, систему команд и режим работы.

**Круг решаемых задач.** Каждый исполнитель создаётся для решения некоторого круга задач — построения цепочек символов, выполнения вычислений, построения рисунков на плоскости и т. д.

**Среда исполнителя.** Область, обстановку, условия, в которых действует исполнитель, принято называть средой данного исполнителя. Исходные данные и результаты любого алгоритма всегда принадлежат среде того исполнителя, для которого предназначен алгоритм. Среду можно рассматривать как полный набор характеристик, описывающих состояние исполнителя.

**Система команд исполнителя.** Совокупность всех команд, которые могут быть выполнены некоторым исполнителем, образует систему команд данного исполнителя (СКИ). Алгоритм составляется с учётом возможностей конкретного исполнителя, иначе говоря, в системе команд исполнителя, который будет его выполнять.

**Режимы работы исполнителя.** Для большинства исполнителей предусмотрены *режимы непосредственного (ручного) управления и программного управления*. В первом случае исполнитель немедленно выполняет каждую поступившую команду. В таком режиме работает пульт кондиционера или телевизора. Во втором случае исполнителю сначала задаётся полная последовательность команд (программа), а затем он выполняет все эти команды в автоматическом режиме. Например, в память стиральной машины

заложены разные достаточно сложные программы стирки, каждая из которых предполагает ряд последовательных действий.

Рассмотрим примеры исполнителей.

### Пример 6

Исполнитель **Черепаша** перемещается на экране компьютера, оставляя след в виде линии. Система команд Черепашки состоит из следующих команд:

Вперёд  $n$  (где  $n$  — целое число) — вызывает передвижение Черепашки на  $n$  шагов в направлении движения — в том направлении, куда развёрнуты её голова и корпус;

Направо  $m$  (где  $m$  — целое число) — вызывает изменение направления движения Черепашки на  $m$  градусов по часовой стрелке.

**Запись**

Повтори  $k$  [**Команда 1** **Команда 2** ... **Команда  $n$** ]

означает, что последовательность команд в скобках повторится  $k$  раз.

Подумайте, какая фигура появится на экране после выполнения Черепашкой следующего алгоритма:

Повтори 12 [Направо 45 Вперёд 20 Направо 45]

### Пример 7

Система команд исполнителя **Вычислитель** состоит из двух команд, которым присвоены номера:

1. вычти 1
2. умножь на 3

Первая из них уменьшает число на 1, вторая увеличивает число в 3 раза. При записи алгоритмов для краткости указываются лишь номера команд.

Например, алгоритм 21212 означает следующую последовательность команд:

умножь на 3  
 вычти 1  
 умножь на 3  
 вычти 1  
 умножь на 3

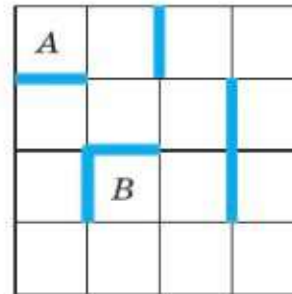
С помощью этого алгоритма число 1 будет преобразовано в 15:  $((1 \cdot 3 - 1) \cdot 3 - 1) \cdot 3 = 15$ .



**Пример 8**

Исполнитель **Робот** действует на клетчатом поле, между соседними клетками которого могут стоять стены. Робот передвигается по клеткам поля и может выполнять следующие команды, которым присвоены номера:

1. вверх
2. вниз
3. вправо
4. влево



При выполнении каждой такой команды Робот перемещается в соседнюю клетку в указанном направлении. Если же в этом направлении между клетками стоит стена, то Робот разрушается.



Что произойдёт с Роботом из примера 8, если он выполнит последовательность команд 32323 (здесь цифры обозначают номера команд), начав движение из клетки *A*? Какую последовательность команд следует выполнить Роботу, чтобы переместиться из клетки *A* в клетку *B*, не разрушившись при этом от столкновения со стеной?

**Пример 9**

К пятизначному натуральному числу применяется следующий алгоритм:

1. Вычислить сумму первых трёх цифр.
2. Вычислить сумму последних двух цифр.
3. Записать полученные два числа друг за другом в порядке возрастания (неубывания).

Пример работы алгоритма для числа 56789:

1.  $5 + 6 + 7 = 18$ .
2.  $8 + 9 = 17$ .
3. Упорядочив, получаем 1718.

Выясним наименьшее и наибольшее пятизначные числа, в результате применения к которым этого алгоритма получается такой же результат.



В старших разрядах наименьшего пятизначного числа должны быть самые маленькие цифры из возможных; первая цифра

должна быть как можно меньше и т. д. В нашем случае это:  $17 = 1 + 7 + 9$ .

Есть единственный вариант, позволяющий получить вторую сумму из двух цифр:  $18 = 9 + 9$ .

Составим наименьшее пятизначное число: 17999.

В старших разрядах наибольшего пятизначного числа должны быть самые большие цифры; первая цифра должна быть как можно больше и т. д. В нашем случае это  $18 = 9 + 9 + 0$ .

Вторую сумму из двух цифр можно получить только так:  $17 = 9 + 8$ .

Составим наибольшее пятизначное число: 99098.

При разработке алгоритма:

- 1) выделяются фигурирующие в задаче объекты, устанавливаются свойства объектов, отношения между объектами и возможные действия с объектами;
- 2) определяются исходные данные и требуемый результат;
- 3) определяется последовательность действий исполнителя, обеспечивающая переход от исходных данных к результату;
- 4) последовательность действий записывается с помощью команд, входящих в систему команд исполнителя.

Можно сказать, что алгоритм — план управления исполнителем.

По ссылке <http://gotourl.ru/11950> вы можете скачать систему программирования КуМир (Комплект учебных Миров) со встроенными исполнителями Робот, Чертёжник, Водолей и др. КуМир работает в операционных системах Windows и Linux.

www

### 3.1.3. Свойства алгоритма

Не любая инструкция, последовательность предписаний или план действий может считаться алгоритмом. Каждый алгоритм обязательно обладает следующими свойствами: дискретностью, понятностью, определённой, результативностью и массовостью.

**Свойство дискретности** означает, что путь решения задачи разделён на отдельные шаги (действия). Каждому действию соответствует предписание (команда). Только выполнив одну команду, исполнитель может приступить к выполнению следующей команды.

**Свойство понятности** означает, что алгоритм состоит только из команд, входящих в систему команд исполнителя, т. е. из таких команд, которые исполнитель может воспринять и по которым может выполнить требуемые действия.

**Свойство определённости** означает, что в алгоритме нет команд, смысл которых может быть истолкован исполнителем неоднозначно; недопустимы ситуации, когда после выполнения очередной команды исполнителю неясно, какую команду выполнять следующей. Благодаря этому результат алгоритма однозначно определяется набором исходных данных: если алгоритм несколько раз применяется к одному и тому же набору исходных данных, то на выходе всегда получается один и тот же результат.

**Свойство результативности** означает, что алгоритм должен обеспечивать получение результата после конечного, возможно очень большого, числа шагов. При этом результатом считается не только обусловленный постановкой задачи ответ, но и вывод о невозможности по какой-либо причине продолжения решения данной задачи.

**Свойство массовости** означает, что алгоритм должен обеспечивать возможность его применения для решения любой задачи из некоторого класса задач. Например, алгоритм нахождения корней квадратного уравнения должен быть применим к любому квадратному уравнению, алгоритм перехода улицы должен быть применим в любом месте улицы, алгоритм приготовления лекарства должен быть применим для приготовления любого его количества и т. д.

### Пример 10

Рассмотрим один из методов нахождения всех простых чисел, не превышающих некоторое натуральное число  $n$ . Этот метод называется «решето Эратосфена» по имени предложившего его древнегреческого учёного Эратосфена (III в. до н. э.).

Для нахождения всех простых чисел, не больших заданного числа  $n$ , следуя методу Эратосфена, нужно выполнить такие шаги:

1. Выписать подряд все натуральные числа от 2 до  $n$  (2, 3, 4, ...,  $n$ ).
2. ЗаклЮчить в рамку 2 — первое простое число.
3. Вычеркнуть из списка все числа, делящиеся на последнее найденное простое число.



4. Найти первое неотмеченное число (отмеченные числа — зачёркнутые числа или числа, заключённые в рамку) и заключить его в рамку — это будет очередное простое число.
5. Повторять шаги 3 и 4 до тех пор, пока не останется неотмеченных чисел.

Рассмотренная последовательность действий является алгоритмом, так как она удовлетворяет свойствам:

- дискретности — процесс нахождения простых чисел разбит на шаги;
- понятности — каждая команда понятна ученику 8 класса, выполняющему этот алгоритм;
- определённости — каждая команда трактуется и выполняется исполнителем однозначно; имеются указания об очередности выполнения команд;
- результативности — через некоторое число шагов достигается результат;
- массовости — последовательность действий применима для любого натурального  $n$ .

Рассмотренные свойства алгоритма позволяют дать более точное определение алгоритма.

**Алгоритм** — это предназначенное для конкретного исполнителя описание последовательности действий, приводящих от исходных данных к требуемому результату, которое обладает свойствами дискретности, понятности, определённости, результативности и массовости.



### 3.1.4. Возможность автоматизации деятельности человека

Разработка алгоритма, как правило, трудоёмкая задача, требующая от человека глубоких знаний, изобретательности и больших временных затрат.

Решение задачи по готовому алгоритму требует от исполнителя только строгого следования заданным предписаниям.

#### Пример 11

Из кучи, содержащей любое, большее 3, количество каких-либо предметов, двое играющих по очереди берут по одному или по два предмета. Выигрывает тот, кто своим очередным ходом сможет забрать все оставшиеся предметы.



Рассмотрим алгоритм, следуя которому первый игрок наверняка обеспечит себе выигрыш.

1. Если число предметов в куче кратно 3, то уступить ход противнику, иначе начать игру, взяв 1 или 2 предмета так, чтобы осталось количество предметов, кратное 3.
2. Своим очередным ходом каждый раз дополнять число предметов, взятых соперником, до 3 (число оставшихся предметов должно быть кратно 3).



Предложите сыграть в эту игру кому-нибудь из своих родных, друзей или знакомых. Действуйте строго в соответствии с приведённым выше алгоритмом. Убедитесь, что алгоритм «работает»!

Исполнитель может не вникать в смысл того, что он делает, и не рассуждать, почему он поступает так, а не иначе, т. е. он может действовать формально. Способность исполнителя действовать формально обеспечивает возможность автоматизации деятельности человека. Для этого:

- 1) процесс решения задачи представляется в виде последовательности простейших операций;
- 2) создаётся машина (автоматическое устройство), способная выполнять эти операции в последовательности, заданной в алгоритме;
- 3) человек освобождается от рутинной деятельности, выполнение алгоритма поручается автоматическому устройству.

Мир, окружающий современного человека, с каждым днём наполняется всё более совершенными формальными исполнителями (цифровыми устройствами), «берущими» на себя многие рутинные виды деятельности, ранее выполнявшиеся человеком. Представления об алгоритмах позволяют понять, как работают цифровые устройства, какие ещё работы можно на них переложить, с какими трудностями при этом можно столкнуться.

### САМОЕ ГЛАВНОЕ

Исполнитель — некоторый объект (человек, животное, техническое устройство), способный выполнять определённый набор команд.

Формальный исполнитель одну и ту же команду всегда выполняет одинаково. Для каждого формального исполнителя можно указать: круг решаемых задач, среду, систему команд и режим работы.

Алгоритм — предназначенное для конкретного исполнителя описание последовательности действий, приводящих от исходных данных к требуемому результату, которое обладает свойствами дискретности, понятности, определённости, результативности и массовости.

Способность исполнителя действовать формально обеспечивает возможность автоматизации деятельности человека.

### Вопросы и задания

1. Что называют алгоритмом?
2. С помощью поиска в сети Интернет выясните происхождение термина «алгоритм».
3. Подберите синонимы к слову «предписание».
4. Приведите примеры алгоритмов, изучаемых вами в школе.
5. Кто может быть исполнителем алгоритма?
6. Приведите пример формального исполнителя. Приведите пример, когда человек выступает в роли формального исполнителя.
7. От чего зависит круг решаемых задач исполнителя «компьютер»?
8. Рассмотрите в качестве исполнителя текстовый процессор, имеющийся на вашем компьютере. Охарактеризуйте круг решаемых этим исполнителем задач и его среду.
9. Что такое команда, система команд исполнителя? Какие команды должны быть у робота, выполняющего функции: а) кассира в магазине; б) дворника; в) охранника?  
Обсудите эти вопросы в группе.
10. Исследуйте один из исполнителей системы программирования КуМир. Охарактеризуйте его назначение, среду, СКИ, возможности ручного и программного управления. Используйте встроенную в систему справочную информацию.
11. Перечислите основные свойства алгоритма.
12. К чему может привести отсутствие какого-либо свойства у алгоритма? Приведите примеры.
13. В чём важность возможности формального исполнения алгоритма?
14. Последовательность чисел строится по следующему алгоритму: первые два числа последовательности принимаются равными 1; каждое следующее число последовательности при-



нимается равным сумме двух предыдущих чисел. Запишите 10 первых членов этой последовательности. Выясните, как называется эта последовательность.

15. Некоторый алгоритм получает из одной цепочки символов новую цепочку следующим образом. Сначала записывается исходная цепочка символов, после неё записывается исходная цепочка символов в обратном порядке, затем записывается буква, следующая в русском алфавите за той буквой, которая в исходной цепочке стояла на последнем месте. Если в исходной цепочке на последнем месте стоит буква «Я», то в качестве следующей буквы записывается буква «А». Получившаяся цепочка является результатом работы алгоритма. Например, если исходная цепочка символов была «ДОМ», то результатом работы алгоритма будет цепочка «ДОММОДН». Дана цепочка символов «КОМ». Сколько букв «О» будет в цепочке символов, которая получится, если применить алгоритм к данной цепочке, а затем ещё раз применить алгоритм к результату его работы?
16. Найдите в сети Интернет анимацию шагов алгоритма Эратосфена. С помощью алгоритма Эратосфена найдите все простые числа, не превышающие 50.
17. Что будет результатом исполнения Черепахой (см. пример 6) следующего алгоритма?

Повтори 8 [Направо 45 Вперед 45]

18. Запишите алгоритм для исполнителя Вычислитель (см. пример 7), содержащий не более 5 команд:
- получения из числа 3 числа 16;
  - получения из числа 1 числа 25.
19. Система команд исполнителя Конструктор состоит из двух команд, которым присвоены номера:
- приписать 2
  - разделить на 2

По первой из них к числу приписывается справа 2, по второй число делится на 2.

- Как будет преобразовано число 8, если исполнитель выполнит алгоритм 22212?
- Составьте алгоритм в системе команд этого исполнителя, по которому число 1 будет преобразовано в число 16 (в алгоритме должно быть не более 5 команд).

20. В какой клетке (*A* или *B*) должен находиться исполнитель Робот из примера 8, чтобы после выполнения алгоритма 3241 (где цифры — это номера команд Робота) в неё же и вернуться?
21. К пятизначному натуральному числу применяется следующий алгоритм:
1. Вычислить сумму первых двух цифр.
  2. Вычислить сумму последних трёх цифр.
  3. Записать полученные два числа друг за другом в порядке возрастания (неубывания).
- Выясните наименьшее и наибольшее пятизначные числа, в результате применения к которым этого алгоритма получится число 1215.
22. К четырёхзначному натуральному числу применяется следующий алгоритм:
1. Вычислить сумму первых двух цифр.
  2. Вычислить сумму последних трёх цифр.
  3. Записать полученные два числа друг за другом в порядке возрастания (неубывания).
- Выясните, какие из приведённых ниже чисел могут получиться в результате работы этого алгоритма: 2118, 1818, 1718, 1214, 123.
23. Все алгоритмы, которые мы рассматривали до этого, можно считать алгоритмами последовательными. Подумайте сами почему. Вместе с тем в реальной жизни очень много принципиально иных алгоритмов. Параллельный алгоритм — алгоритм, который может быть реализован по частям на множестве различных исполнителей с последующим объединением полученных результатов и получением корректного результата. Приведите 2–3 примера параллельных алгоритмов из окружающего нас мира.
24. Три актёра готовятся к спектаклю. С ними работают два опытных гримёра. Каждый актёр должен быть покрашен и причёсан. Макияж у каждого актёра продолжается полчаса, а причёсывание — только 10 минут. Спланируйте работу гримёров так, чтобы актёры как можно быстрее подготовились к выходу на сцену. Сколько для этого потребуется времени?
25. Группа из четырёх туристов должна пройти по мосту в темноте. Идти по мосту одновременно могут не более двух туристов. При этом они могут пользоваться только одним



фонарём. Перебросить фонарь с одного берега на другой нельзя, поэтому кто-то из них должен вернуться с фонарём. Аня проходит через мост за 1 минуту, Борис — за 2 минуты, Тимур — за 5 минут и Дана — за 10 минут. Какое наименьшее время требуется туристам, чтобы все они перешли по мосту на другой берег?

## § 3.2

### Способы записи алгоритмов

**Ключевые слова:**

- словесное описание
- Школьный алгоритмический язык
- построчная запись
- псевдокод
- блок-схема

Существуют различные способы записи алгоритмов. Основными среди них являются:

- словесные (на естественных языках);
- графические;
- на языках программирования.



Теоретические исследования нашего соотечественника Андрея Андреевича Маркова (младшего) (1903–1979), выполненные в середине прошлого века, показали, что в общем случае алгоритмы должны содержать предписания двух видов:

- 1) предписания, направленные на непосредственное преобразование информации (функциональные операторы);
  - 2) предписания, определяющие дальнейшее направление действий (логические операторы).
- Именно эти операторы положены в основу большинства способов записи алгоритмов.

#### 3.2.1. Словесные способы записи алгоритма

**Словесное описание.** Самой простой является запись алгоритма в виде набора высказываний на обычном разговорном языке. Словесное описание имеет минимум ограничений и является наименее формализованным. Однако все разговорные языки облада-